Quality Adaptation in P2P TV Based on Scalable Video Coding

Youssef Lahbabi[1], Ahmed Hammouch[2], El Hassan Ibn Elhaj[3]
[1]ENSET Mohammed V Souissi University
[2]Mohammed V Souissi University, ENSET
LRGE
[3]National Institute of Posts and Telecommunications (INPT)
Rabat, Morocco
Lahbabi_youssef@yahoo.fr, hammouch_a@yahoo.com, ibnelhaj@inpt.ac.ma

ABSTRACT: *Recently, Peer-to-Peer (P2P) techniques for Internet Protocol Television (IPTV) have attracted a lot of attention due to their high potential at improving the performance of today's multimedia systems. In this paper, we propose a novel concepts and mechanisms that enable the use of Scalable Video Coding (SVC) in Peer–to–Peer Television (P2PTV) system to achieve the quality adaptation. Using SVC, we have developed a two-stage quality adaptation algorithm that matches the video quality with available local and system resources. Key components with different our design options are presented and experimentally evaluated, with the objective of investigating benefits of network P2P in combination with SVC. We have carried out extensive experiments on real stream data to (i) evaluate the performance of our design in terms of adapted video stream SVC with available resources (ii) more robustness against high churn rates, heterogeneous peers, and flash crowd scenarios. Our results demonstrate the feasibility of the approach and bring us one step closer to real adaptive peer-to-peer streaming.*

Keywords: Peer-to-peer Streaming, Live Streaming, IPTV, P2PTV, Scalable Video Coding

Received: 11 July 2014, Revised 13 August 2014, Accepted 15 August 2014

1. Introduction

P2PTV systems for streaming are becoming more popular in our world since they allow supporting more peers simultaneously, better load distribution, and scalability. Additionally, it enhances the capacity of the P2P network and, thus, either increases the achievable bitrate.

Nonetheless, current P2PTV systems still suffer from a major limitation: such these systems try to provide the single video quality to all users even if they have different devices with a large spectrum of resources. We believe that it is of essence that future P2PTV systems are quality adaptive, meaning that different devices may retrieve different quality layers based on available resources and each user can take more or less layers according to its capabilities.

In recent years, the research community has put substantial effort into investigating adaptive P2P video streaming [1, 2]. This includes different aspects such as theoretical models [3], replication techniques [4], prefetchingpolicies [5], network awareness [6], and the impact of server allocation [7]. It was early recognized that video coding techniques are crucial for a high streaming experience [8, 9]. Streaming video content poses special and usually strict requirements on data transmission. Video files are usually large and require much more bandwidth than audio files. As long as resources are guaranteed, data transmission would proceed as expected. However, the Internet rarely offers guarantees, thus performance is drastically affected upon resource scarcity. Classical approaches [1, 10] started off by being agnostic to the video codec, and researches would resort to standard video codecs that exhibit the highest compression ratios [1]. These codec-agnostic approaches are also known as single layer video systems. Evidently, different parts for the video stream have different impacts on the video if they were to go missing. For example, key video frames are required to decode other prediction video frames. Losing the key frames means that data depending on it cannot be decoded. Adaptation was a main drive for much research in the area of P2P streaming. But before having the availability of advanced multi-layer codecs, researchers had to resort to simply extracting information about the importance of different video blocks from the video itself.

Fortuna et al. [10] use the properties of single-layer video codecs to realize a pull-mesh-based P2P streaming system. This work applies a special piece scheduling algorithm that uses priorities derived from the video blocks and the frames they contain. Therefore, if a block is carrying prediction frames and resources are not sufficient to send it, then the system scales and the additional frames are simply dropped. Although this is considered a basic kind of scalability, it was, nonetheless, an important step forward to more advanced adaptation techniques. As outlined above, it is essential in scenarios with resource heterogeneity that multi-layer video coding techniques are used. This allows operating in the presence of devices with varying characteristics, from desktop computers to handhelds [11]. Furthermore, quality can be switched during playback to adapt to changing network conditions and system load. Early research on quality adaptation started by considering general multilayer codecs with focus on single dimension scalability [12, 13, 14, 15, 16, 17]. Therefore, these systems did make a clear distinction between temporal, spatial, and SNR scalability. Rejaie et al. present PALS [17], a receiver driven P2P video streaming system with quality-adaptive playback of layered video. However, PALS only considers single dimensional scalability which cannot adapt to heterogeneous networks. Baccichet [18] use a prioritization mechanism and multicast trees to distribute SVC streams. Osama Abboud [19] presents quality adaptive P2P streaming providing network status, bitrate and complexity adaption using scalable video coding. But, he can’t adapt a video SVC with highly fluctuating as churn rate and flash crowd situation.

In this paper, we make use of the state of the art in multi-layer coding, namely SVC, to enable quality adaptation in P2PTV systems in order to adapt to various static and dynamic parameters of the peers and the system. Our objective is to provide concepts and algorithms that address the problems of adaptation in a P2PTV system based on SVC. In particular, our results show that our proposed algorithms can achieve the best adaptation of quality level with available resources and contribute to give the more robustness against high churn rates, heterogeneous peers, and flash crowd arrivals of peers.

The remainder of this paper is organized as follows. A brief background on scalable video coding and P2PTV in Section 2.The proposed of our quality adaptation algorithms that use SVC is described in details in Section 3. Section 4 illustrates the simulation results as well as our analysis and we conclude the paper in Section 5.

2. Background

In this section, to start with, we will give a brief introduction to the Peer -to-Peer (P2P) technology and scalable video coding (SVC). P2PTV is a category of P2P programs that are specifically designed for live video streaming.

2.1 Scalable Video Coding (SVC)

Quality adaptation becomes important when a receiver peer select multiple sender peers to receive any video. We focused to use H.264/SVC [20] [21] for encoding the original video contents that are used to produce highly compressed bit streams to generate a wide variety of bit rates. In such video encoding scheme, original video stream is truncated into many different video layers. The “*base layer*” provides a significant proportion of the video quality whereas the “*enhancement layers*” are used to enhance the video quality in different dimensions. “*Base laye*r” is considered as the most important layer because all the “*enhancement layers*” are only decodable with reference to lower layers and “*base layer*”. Initially the source peer distributes the “*base layer*” among all super peers. The enhancement layers are exchanged among the clusters using the flat hierarchical structure of the super peers. The packet ordering based quality adaptation not only ensures the base video quality for streaming session but also adds resilience for the lower video layers. The SVC based video coding offers the receiver to select the video either with high SNR quality, temporal resolutions for varying frame rates, or spatial resolution for the different resolutions (Figure1).

2.2 Peer-to-Peer Television (P2PTV)

Streaming media refers to the continuous audio/video streams that are transmitted and played over network with regard to also adds resilience for the lower video layers. The SVC based video coding offers the receiver to select the video either with high SNR quality, temporal resolutions for varying frame rates, or spatial resolution for the different resolutions (Figure1).

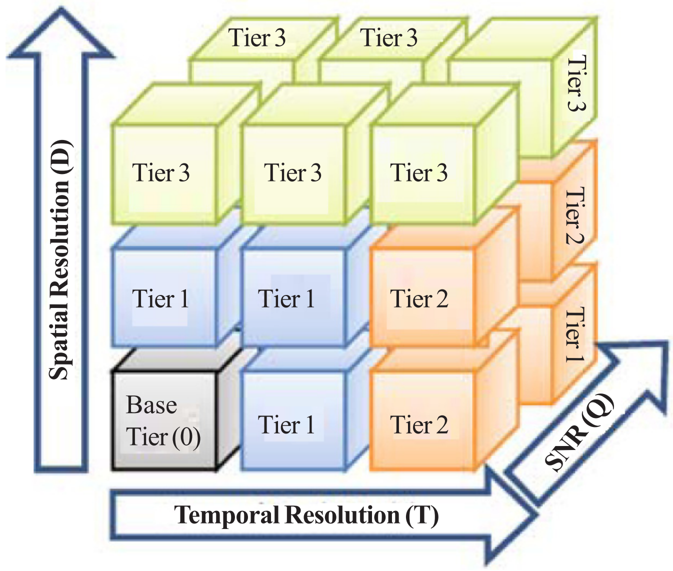

Figure 1. 3D representation of an SVC bistrea

time sequence. Streaming media has three haracteristics: Continuous, Real-time, and Time Sequential. P2PTV is a category of P2P software applications specially designed to redistribute video streaming media based on a P2P network. Compared to P2P software designed for file sharing which has few concerns of timeliness property, P2PTV is born for real-time contents. Typically, the broadcasted video streams are channels from all over the world but may also come from other sources, e.g. recorded video files. The draw to these applications is significant because they have the potential to make any TV channel globally available. Based on the type of distribution topology graph, P2PTV systems can be categorized as BitTorrent-like mesh based [22] and Treebased. A BitTorrent-like P2PTV system can be viewed as a real-time version of BitTorrent [23]. At the content provider, the video stream is split into chunks. These chunks are distributed, or transmitted to different peers. A tracker server keeps and updates the chunk buffer map, which bears the information of which peer has which chunk. A newly joined peer consults the tracker server for the chunk buffer map and then based on this information it downloads the current required chunks from the corresponding peers. The overlay of the peers is not in a hierarchical but an ad hoc structure. Peers with more resources are placed close to the content provider (source) for better performance. Peers will move further/closer to the source according to the changes of the network conditions. By doing so, these systems can accommodate peers with different resources and in particular with different access capacity. BitTorrent-like P2PTV system inherits the concept of swarm from BitTorrent-protocol: A peer connects to other peers to obtain the various chunks; such a group of peers connected to each other for a same video stream is called a swarm of P2PTV. In BitTorrent file sharing system, a swarm does not necessarily contain the initial content provider (seeder), however for BitTorrent-like P2PTV systems, it is always the case: otherwise it means the content provider has left the network, which is, in other words, no more updates for the real-time video stream: the channel turns offline.

If the swarm contains only the content provider, the client peer connects directly to it and begins to request chunks. Instead of downloading only directly from the content provider, peers begin to trade chunks with each other when they enter the swarm. Similar to BitTorrent file sharing systems, mechanisms that encourage fairness, are also implementable for BitTorrentlike P2PTV systems. As the most widely used BitTorrent-like P2PTV systems, SopCast [24], PPLive [25], PPStream [26], UUsee [27] are considered to be the most typical examples.

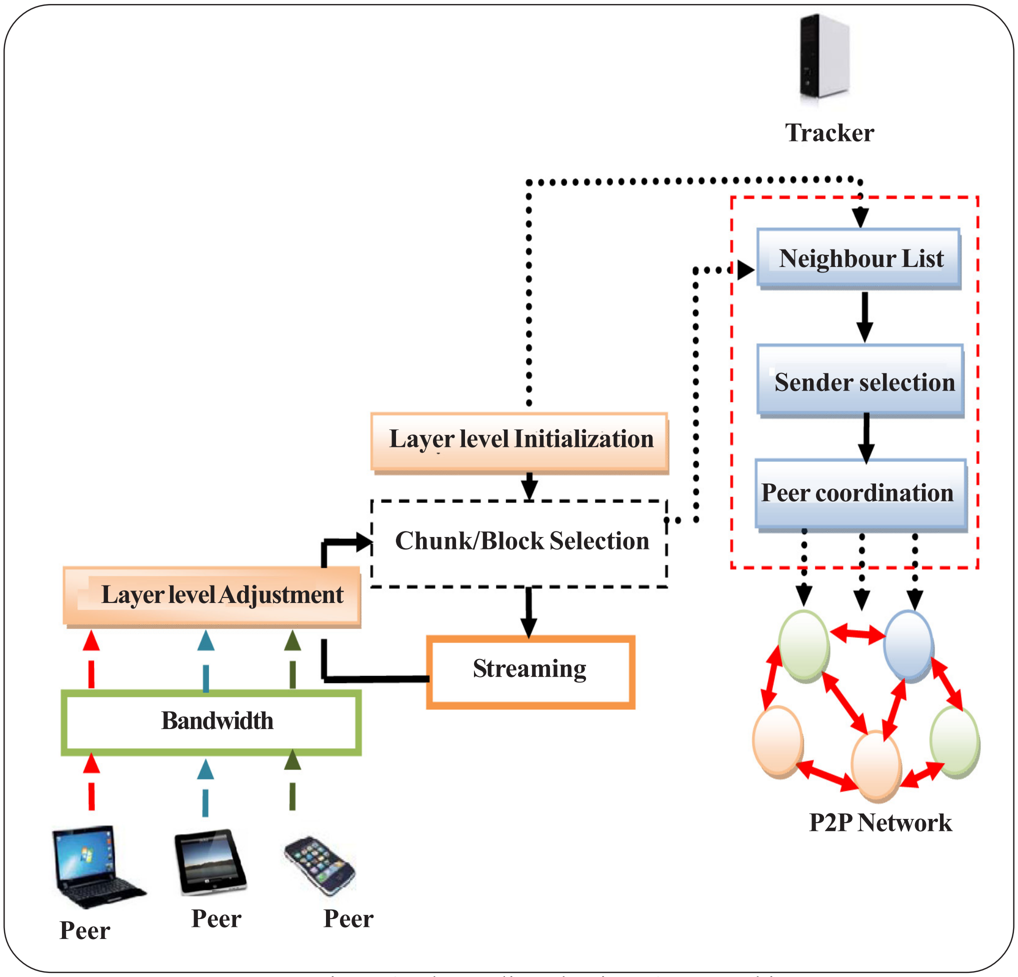

Figure 2. The quality adaptive P2PTV architecture

3. Proposed Quality Adaptive Streaming

In this section, we describe our SVC-based qualityadaptive P2PTV system. We assume in our system model, there are three entities: tracker, source, and peer. The tracker matches peers who are viewing the same video stream. This matching results in multiple dynamic swarms in the system. There is at least one source node in the system to introduce the original streams to peers. The source node (sometimes called seed server) also provides additional capacity in case that peers do not have enough resources and in the beginning of the sessions where very few peers exist. Figure 2 depicts the basic architecture of our quality adaptation in P2P-TV system.

Quality adaptation is achieved by adjusting quality according to the different peer resources and network dynamics. It is performed by two modules: the Layer Level Initialization (LLI) and the Layer Level Adjustment (LLA). Both modules form the algorithms that match the layers with resources available at the peer. On the one hand, the LLI is ed for determining the highest possible layer that a peer can retrieve and play, and is performed at session start. On the other hand, the LLA is performed periodically to adjust the layer according to the changes of the network environment. After the playback is initiated, the LLI is first called to make a decision on the feasible quality level based on local resources. Based on this decision, peer selection and block selection are performed. Peers are selected in such a way that they are able to provide the selected layer. After the neighboring peers have been contacted and upload slots have been reserved, block selection is done. To ensure continuous playback, the LLA is performed regularly, and if required, it may increase or decrease the selected layer accordingly. Next, we give the details of the quality adaptation modules and their role in the P2PTV system.

3.1 Layer Level Initialization (LLI)

Layer Level Initialization (LLI) is typically invoked only once at the beginning of the playback session. It is designed in such way that each peer can determine its highest SVC quality level before starting to stream the SVC video. The main idea of LLI is to compare the requirements of each layer of the video stream with the local static resources of a peer. The subtle property of the LLI is that it has to make a decision on the quality level without having any information about system throughput and dynamics. The LLI module evaluates the current resources and requirements in order to match them with achievable quality. This module mainly handles static parameters, such as screen resolution, bandwidth, currently available device power (CPU, RAM, Battery Life), and user preference (Display Resolution, Frame Rate, PSNR Level). An initial quality set with base layer quality level parameters d_0 , t_0 and q_0 is populated at first. And then, the spatial calability, temporal scalability, bitrate [28], quality scalability, complexity adaptation [29] and distributions video length relative battery Life [30] [31] adaptation modules select out all compatible quality level based on screen resolution, bandwidth, and device power respectively considering the user preference limitations. All the compatible combinations are appended as candidates. The final decision is made by selecting the filtered quality set $L_1^{d,t,q}$ which values of all three dimensions are at their maximum. The LLI final decision is prefer items with higher SNR value. The proposed quality level initialization algorithm is shown in Algorithm 1.

Algorithm 1: Layer Level Initialization algorithm (LLI)

Input:
$d_i = d_0$; $t_i = t_0$; $q_k = q_0$; d_0, t_0 , q_0 represent initial three types of scalability .
d_a, t_a ,q_a; represent adaptive three types of scalability.
β; Frame Rate adaptive.
μ ; PSNR Level adaptive.
φ ; Quantization Stepsize.
Initial layer level set candidate CandidateDTQ $\{d_i , t_j , q_k\}$
α; Distributions Video length relative Battery Life.
Output: Layer level adaptive $L_1^{d,t,q}$ suited to the static peer resources limitation.
Begin
If (($d_i \le d_a$) and (UserPerferences.DisplayResolution ≤ PeerResources Screen. Resolution)) then
  If ($t_j \le t_a$) and (UserPreferences.FrameRated ≤ β) and (Bitrate (β, φ) ≤ PeerResources. Bandwidth)) then
    If (($q_k \le q_a$) and (UserPreferences.PSNRLevel ≤ μ) and (Complexity (d_i, t_j , q_k) ≤ PeerResources. Device Power) and ($\alpha \le$ 1))
      CandidateDTQ.append $\{d_i , t_j , q_k\}$;
End If
 $q_k = q_k + 1$;
End If
    $t_j = t_j + 1$;
  End If
  $d_i = d_i + 1$;
  Return Maximize CandidateDTQ.append $\{d_i , t_j , q_k\}$
Else
 Return "*Error* : *static resources too low for base stream*"
End

3.2 Chunk / Block Selection

In our architecture, each peer maintains a set of peers $\pi = \{ p_1, p_2, ..., p_n\}$ serving the video stream. Each peer is divided into uploders $U = \{ u_1, u_2,\ldots, u_l \}$ or downloaders $D = \{d_1, d_2, \ldots, d_k\}$ and the video file is divided into chunks. Based on the SVC design, each video chunk is further divided into blocks. Each Block is described by its serial number sn_j in the overall stream, its size s_j , and the index of the layer l_j it belongs to $b_j = (sn_j , s_j , l_j)$. Block selection (Figure 3) is an important part of our streaming architecture, since it is behind making a decision on which blocks to request. When a peer connects to the overlay to begin downloading the stream, it exchanges a buffer map that indicates the availability of the Blocks described in the stream map . Considering a stream that contains M Blocks, and then the Buffer Map is as follows in equation 1:

$$\{B_j, 1 \le j \le M\}; BM_j = \begin{cases} 1 & \textit{if the Block is available} \\ 0 & \textit{else} \end{cases} \tag{1}$$

All the received buffer maps from the peers that belong to the same overlay will form a matrix $(BM_{ij})_{1 \le i \le N;\, 1 \le j \le M}$(where BM_{ij} indicates the availability of the Block b_j at the peer p_i).

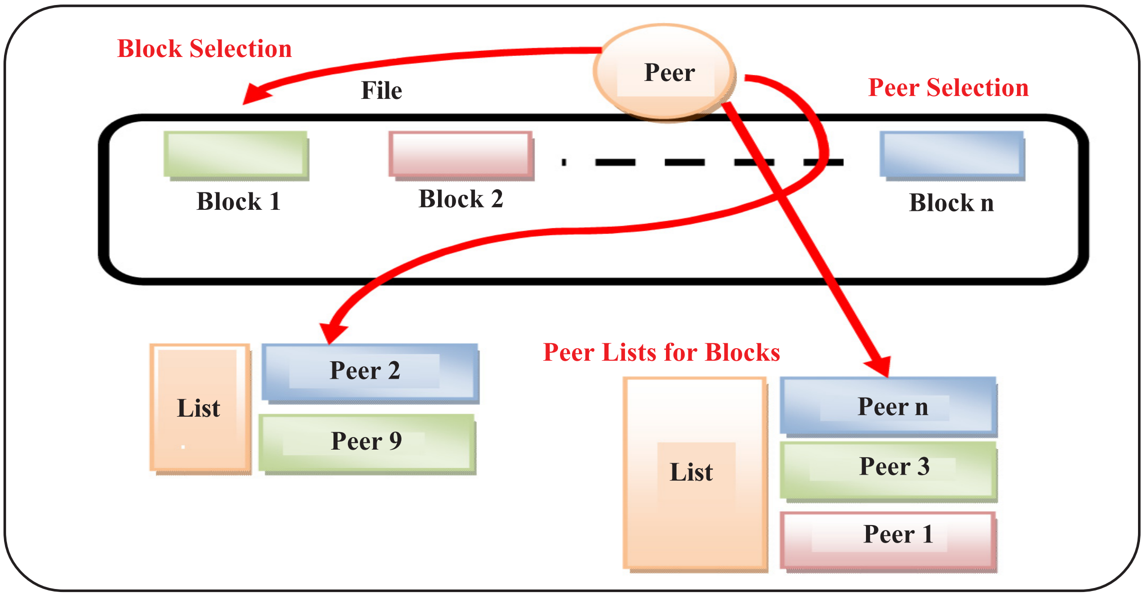

Figure 3. Chunk/ Block Selction scenario

3.3 Layer Level Adjustment (LLA)

The LLA is executed periodically while streaming as a part of a control loop to ensure smooth adaptation. This module adapts to changes in network conditions in order to maximize available quality at the receiver. Since the screen size of the user does not change during the video playback, only temporal and SNR adaptations are handled by the LLA because typically the peer display resolution is unchangeable. The spatial layer will not be changed by the LLA. The LLA uses real-time information of network status measured through the block availability in the partnership size and the active download throughput. It takes the current SVC layer adaptive as input, then adjusts it according to the real time network information. This layer is processed by the different stages of the LLA to produce a new layer that fits the current network conditions. Therefore, the LLA starts from the pre-filtered of LLI output layer level parameters $\{d, t_t, q_t\}$. The five adaptation stages of the LLA form together the decisionmaking process. The network status, stream rate, bitrate, complexity, and PSNR level adaptation components adjust all compatible quality level based on block availability, churn rate, throughput , and device power (CPU, RAM, Battery life) respectively. Here, the block availability indicator provides information about layers that are available in the P2PTV system. The proposed layer level adjustment algorithm is shown in Algorithm 2.

3.1.1 Streaming Rate

The average streaming rate R [32] is given by:

$$R = \left(u_p + \frac{u_s}{N}\right) \tag{2}$$

Where N is the number of all active peers in the system. We use U_p to denote the average upload capacity participating peers and U_s to denote the upload capacity of dedicated streaming server.

3.1.2 Churn Rate

Algorithm 2 : Layer level Adjustment (LLA)

Input:
$d_i = d$; Represents Spatial scalability constant in real time.
$t_j = t_t$; Represents Temporal scalability varying in real time.
$q_k = q_t$; Represents Quality scalability varying in real time.
Current layer level set candidateDTQ $\{d, d_j, q_k\}$.
t_a; Represents adaptive temporal of scalability.
q_a; Represents adaptive quality of scalability.
β ; FrameRate adaptive.
μ; PSNR Level adaptive.
φ;Quantization Stepsize.
Output: Layer level adaptive $L_1^{d,t,q}$ suited to the real times peer resources and device power limitation.
Begin
Test and filtered three types of scalability (d, t_j, q_k); test block availability.
CandidateDTQ.append $\{d, t_j, q_k\}$;
Adapt VideoSVC.StreamRate **with**
Real Time Information.ChurnRate; Calculate R.
If $((t_j \leq t_a)$ and (VideoSVC.FrameRated $\leq \beta$) and (Bitrate $(\beta, \varphi) \leq$ Real Time Information.Throughput)) then
If $((q_k \leq q_a)$ and (VideoSVC.PSNRLevel $\leq \mu$) and (Complexity $(d, t_j, q_k) \leq$ Peer Resources. Device Power)
CandidateDTQ.append $\{d, t_j, q_k\}$;
$q_k = q_k + 1$;
End If
$t_j = t_j + 1$;
End If
Return Maximize CandidateDTQ.append $\{d, t_j, q_k\}$
Return "*Error :real times resources too low for base stream*"
End

We refer to the ratio of the total number of peers λ that join the streaming system during the simulation time to the total number μ of peers that leave the system as the churn rate ρ [33].

$$\rho = \frac{\lambda}{\mu} \tag{3}$$

4. Simulation

4.1 Experimental Setup

We have implemented the proposed quality adaptive streaming in simulator PSIM [34] by Java language. Our implementation was validated by using actual video stream. To conduct rigorous quantitative analysis of the proposed algorithms under wide range of working conditions, we implemented a testing application to emulate the characteristics of realistic P2PTV systems. This testing application enables us to conduct controllable and repeatable experiments with different parameters and large number of peers. The setup of our experiments is as follows. Our simulation lasts for 10 min with varied cross traffic to present the dynamic end-to-end resources. We assume that all arrivals and departures are scheduled according to a Poisson distribution during time. Each peer can join the system at different time, and will depart after its lifetime. We create a highly-dynamic P2P streaming system with 180 heterogeneous peers that are continually changing and we consider having one server with 4086 Kbps. The basic setup used for the performance evaluation is shown in Table 1. The upload bandwidth values of peers are

Parameter	Value
Simulation duration	10 minutes
Number of peers	180
Peer arrival pattern	Poisson
Number of servers	1
Server upload capacity	4086 Kbps
Video length	100 frames

Table 1. Simulation setup

	Set 1	Set 2	Set 3
Number	60	60	60
Screen size	176 ×144	352 × 288	704 × 576
Upload speed	128 Kbps	320 Kbps	800 Kbps
Download speed	256 kps	560 Kbps	1200 Kbps

Table 2. Resource configuration for the peers

chosen according to the distribution given in Table 2. Peers in our system can randomly fail, and they join/leave the system following by Poisson distribution, where this probability distribution is chosen to create a specific testing scenario such as lash crowd arrivals and high peer churn rates. Without losing generality, we consider one video source [35] with length of 100 frames. By using SVC [36], the video source is encoded into a total of 17 layers with one base layer and 16 enhancement layers. The main configuration parameters related to the quality scalability used in this paper are presented in Table 3.

4.2 Exprimental Results

Now, we evaluate how our proposed adaptation algorithms improve the performance of the P2P TV system and we simulate changing parameters to see how the LLA reacts to them. Moreover, we analyze the impact of several system parameters on the performance and robustness of our mechanisms, especially in presence of heterogeneous peers, high peer churn rates and flash crowd arrivals.

4.2.1 Quality adaptive by LLA

Here, we present a preliminary evaluation of our proposed quality adaptation mechanisms. We suppose that the LLI has already decided on a basic spatial level, i.e. $d = 0$ or $d = 1$. From Figure 4, we can see that as the adaptation of decision on SVC level (D, T, Q) by using the LLA module.

4.2.2 Impact of churn rate on layer SVC quality

In this scenario, we run the experiment for the video stream in Table 3 and we enable the module LLA for adapt the real time resources. We measure the average streaming rate during live streaming sessions and we define as the total amount of received video data per second. The average streaming rate is computed across all active peers and represents a basic performance metric. In this evaluation we study the impact of the churn rate on the quality layer level during streaming and we consider a highly dynamic peer-to-peer network with frequent arrivals and departures of peers. In highly dynamic peer-to-peer systems, some peers join the system, start streaming and also contribute their resources to others. At the same time, other peers may be leaving the system, which will result in loss of upload resources and perhaps disruption of some on-going streaming sessions. In our experiments, we control the peer churn rate by tuning the arrival rate l. As l increases, the mean churn rate r also increases as formula (3) where μ is fixed to be 10. As the churn rate increases, the network becomes more dynamic. We measure the layer level perceived across all peers for each churn rate. Figure 5 shows that the variation of churn rate between 1 and 10. From Figures 6 and 7 we can see that the adaptation of average streaming rate and layer level by LLA module with churn rate. When the churn rate increases we observe that the average streaming rate and the number of layer changes becomes fewer from 747.8 (kbit/sec) to 74.78 (kbit/sec) and 17 (SVC Level) to 1 (SVC Level) simultaneously, and vice versa.

Configuration File	Parameter	Value
Main.cfg	FrameRate FramesToBeEncoded GOPSize NumLayers	30 100 16 4
layer0.cfg	SourceWidth SourceHeight FrameRateIn FrameRateOut QP	176 144 15 15 34
Layer1.cfg	SourceWidth SourceHeight FrameRateIn FrameRateOut QP	176 144 15 15 28
Layer2.cfg	SourceWidth SourceHeight FrameRateIn FrameRateOut QP	352 288 30 30 36
Layer3.cfg	SourceWidth SourceHeight FrameRateIn FrameRateOut QP	352 288 30 30 30

Table 3. Main configuration parameters used in the simulation

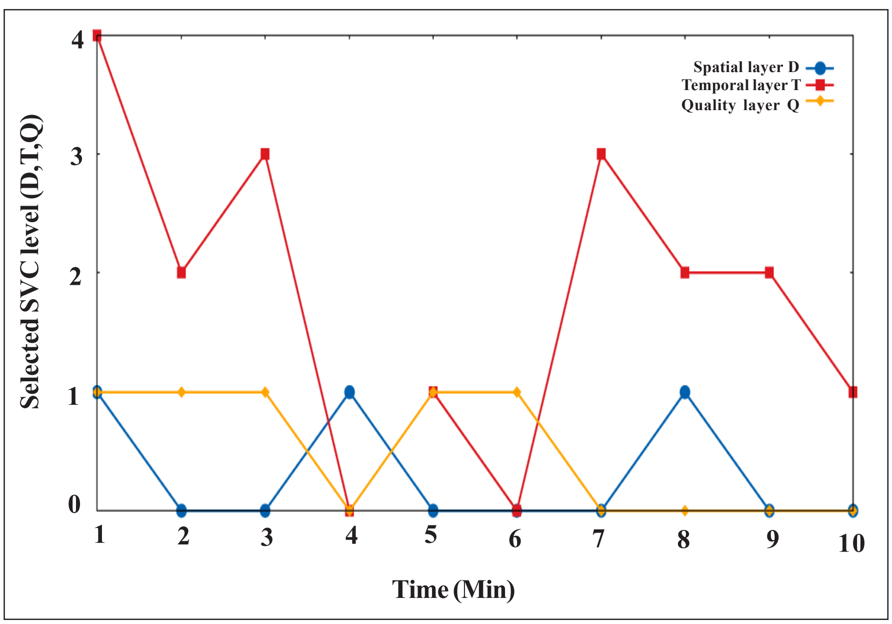

Figure 4. Decision output of the LLA module

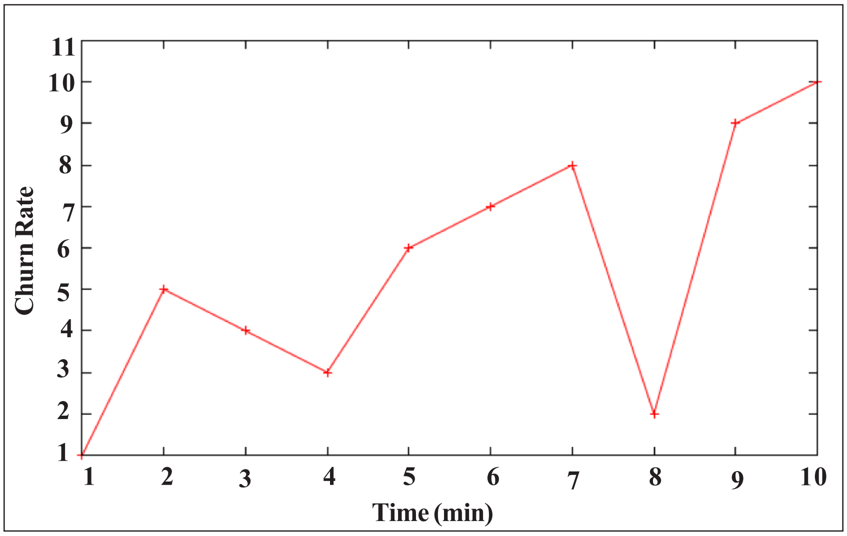

Figure 5. Churn Rate of P2PTV system

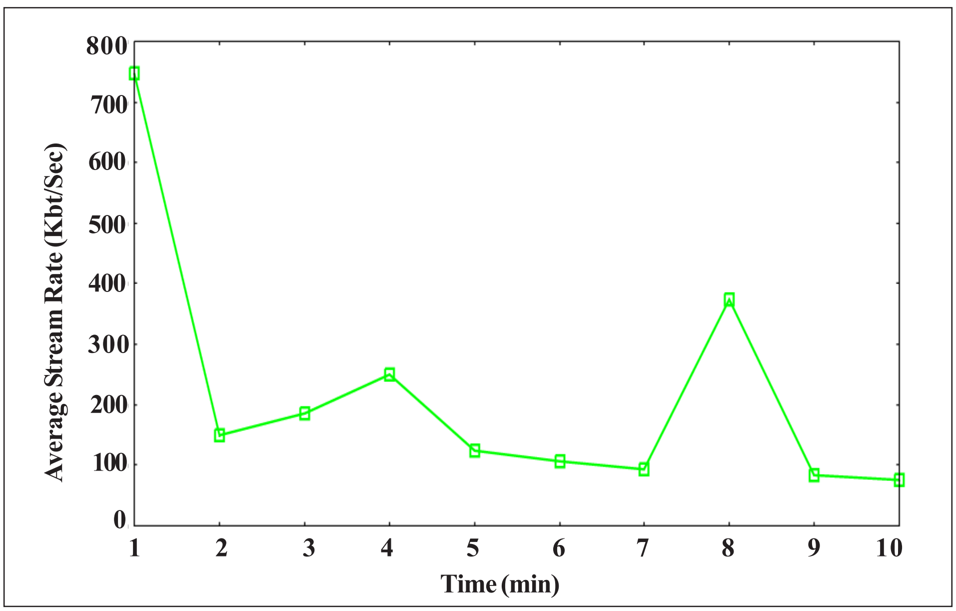

Figure 6. Average Streaming Rate adaptation by LLA module

4.2.3 Impact of flash crowd arrivals

In flash crowd arrivals, peers join the network in a short period of time. In this case, the demand for receiving the video streaming may become more than the available resources. Flash crowds scenarios put a substantial stress on the P2PTV system that strive to provide a reasonable and sustained video quality to peers. We change the average number of peer arrivals per minute from 10 to 100 with an increment of 10. Peers arrive uniformly at random during the simulation period and we measure the SVC bitrate of layer level for considered P2PTV system for each peer arrival rate. Figure 8 demonstrates that while under very high peer arrival rates the SVC bitrate rendered by this system decreases because of the limited upload capacity. This exhibit that our proposed LLA algorithm is can adapt faster the SVC bitrate with an increasing peer arrival rates and is resilient to the impact of flash crowd.

The results show that our mechanisms are able to quickly react to different changes in the P2PTV system while providing best quality that matches current resources and peer dynamics.

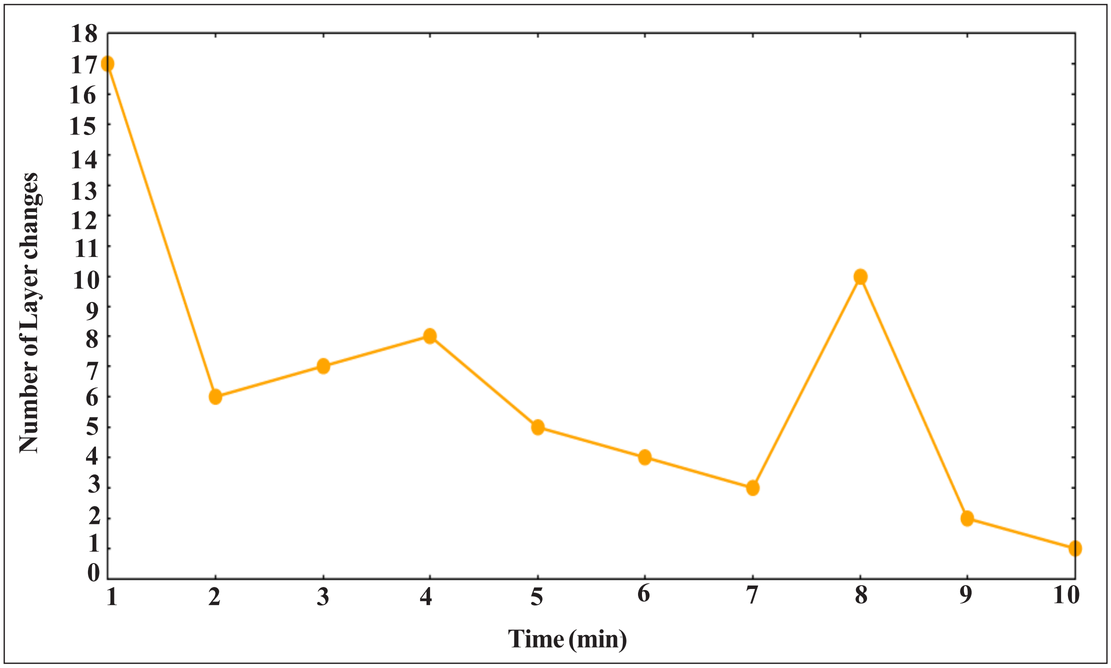

Figure 7. Layer level adaptation by LLA module

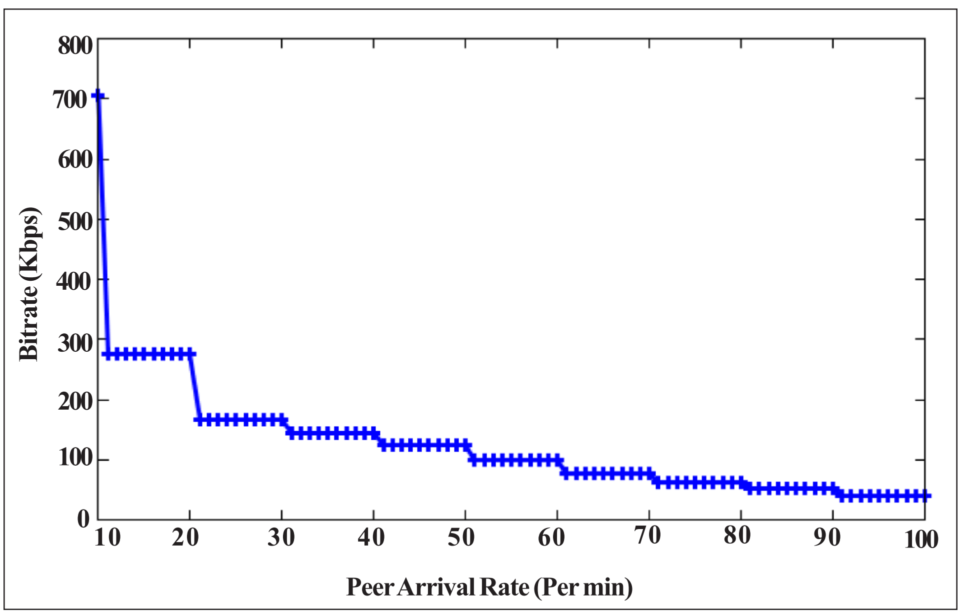

Figure 8. Adaptation of SVC bitrate with the impact of flash crowd by LLA module

5. Conclusion

In this paper, we have introduced a new quality adaptation algorithms used to adapt the video quality to various static and dynamic parameters of the peers and the P2PTV system. The simulation results showed that our mechanisms react quickly to various system changes while providing best quality adaptation of scalable streaming with available resources and unpredictable network. Furthermore, we have demonstrated the effectiveness of the algorithms of the proposed architecture in order to combat the increasingly flash crowd and churn rate environment. Finally, we consider that the use of our quality adaptation is the key for next generation of P2PTV system that is characterized by a highly flexible live streaming.

References

[1] Liu, Y., Guo, Y., Liang, C. (2008). A Survey on Peer-to-peer Video Streaming Systems, *Peer-to-Peer Networking and Applications*, 1 (1) 18–28.

[2] Nguyen, A. T., Li, B., Eliassen, B. (2010) Chameleon: Adaptive peer-to-peer streaming with network coding, *In*: Infocom.

[3] Kumar,R., Liu,Y., Ross, K. W. (2007). Stochastic Fluid Theory for P2P Streaming Systems. *In:* 26th IEEE International Conference on Computer Communications (INFOCOM).

[4] Cheng, B., Stein, L., Jin, H., Zhang, Z. (2008). A Framework for Lazy Replication in P2P VoD. In 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV.

[5] Huang, C., Li, J,. Ross, K. W. (2007). Can Internet Video-on-Demand be Profitable? *In:* ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM).

[6] Hefeeda, M., Bhargava, B.,Yau, D. (2004). A Hybrid Architecture for Costeffective On-demand Media Streaming. *Computer Networks*, 44 (3) 353- 382.

[7] Pussep, K., Abboud,O., Gerlach, F., Steinmetz,R., Strufe,T. (2010). Adaptive Server Allocation for Peer-assisted VoD. In International Parallel and Distributed Processing Symposium (IPDPS).

[8]Annapureddy, S., Gkantsidis, C., Rodriguez, P. L. (2006). Providing Videoon- Demand using Peer-to-Peer Networks. In Proc. of the Internet Protoco TeleVision (IPTV) workshop.

[9] Chi, H., Zhang, Q., Jia, J., Shen, X. (2007). Efficient Search and Scheduling in P2P-based Media-on-Demand Streaming Service. *IEEE Journal on Selected Areas in Communications*, 25 (1)119–130.

[10] Fortuna, R., Leonardi, E., Mellia,M., Meo, M., Traverso, S. (2010). QoE in Pull Based P2P-TV Systems: Overlay Topology Design Tradeoffs. *In:* Proceedings of the 10th International Conference on Peer-to-Peer Computing (P2P).

[11] Cui,Y., Nahrstedt, K. (2003). Layered Peer-to-Peer Streaming. In 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV).

[12] Karlsson, J., Li, H. (2006). P2P Video Multicast for Wireless Mobile Clients.In The Fourth International Conference on MobiSys.

[13] Magharei, N., Rejaie, R. (2006). Adaptive Receiver-driven Streaming from Multiple Senders. *Multimedia Systems*, 11:1–18.

[14] Nguyen, B., Li, Eliassen, F., (2010). Chameleon: Adaptive Peer-to-Peer Streaming with Network Coding. *In:* IEEE INFOCOM.

[15] Liu.,J. Li.,B. Zhang.,Y. -Q.(2003). Adaptive Video Multicast over the Internet. *IEEE MultiMedia*, 10 (1).

[16] Padmanabhan,V. N., Wang,H. J., Chou, P. A. (2004). Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming. *In:* International Workshop on Peer-to-Peer Systems (IPTPS).

[17] Rejaie, R., Ortega,A. (2003). PALS: Peer-to-Peer Adaptive Layered Streaming.*In:* ACM NOSSDAV.

[18] Baccichet,P., Schierl, T,, Wiegand, T., Girod, B. (2007). Low-delay Peer-to-Peer Streaming using Scalable Video Coding, Packet Video.

[19] Osama AbboudB, Konstantin Pussep, Aleksandra Kovacevic, Ralf Steinmetz. (2009). Quality Adaptive Peer-to-Peer Streaming Using Scalable Video Coding. MMNS .

[20] Video sequence SVC, http://media.xiph.org/video/derf/, 2014.

[21] Mederic Blestel and Mickael Raulet. Open svc decoder. Available at http://sourceforge.net/projects/opensvcdecoder/, 2014.

[22] Lu,Y., Mol,J. D. D., Kuipers, F. A., Van Mieghem, P. (2008). Analytical Model for Mesh-based P2PVoD, to appear in Proc. of the 10th IEEE International Symposium on Multimedia (ISM2008), Berkeley, California, USA, December 15-17.

[23] BitTorrent http://www.bittorrent.com, 2014.

[24] SopCast http://www.sopcast.com, 2014.

[25] PPLive http://www.pplive.com/en.index.html, 2014.

[26] PPStream http://www.ppstream.com, 2014.

[27] UUSee http://www.uusee.com, 2014.

[28] Zhan May, Felix C. A. Fernandesy, Yao Wang. (2012). Power-Rate-Quality Optimized Scalable Video Adaptation, *In*: Proc. of IEEE CISS, Princeton NJ, March.

[29] Ma, Z., Wang, Y. (2008). Complexity Modeling of Scalable Video Decoding, in Acoustics, Speech and Signal Processing, ICASSP, p.1125 - 1128, March 31-April 4.

[30]Youssef Lahbabi, El Hassan Ibn Elhaj, Ahmed Hammouch. (2014). Quality adaptation using Scalable Video Coding (SVC) in Peer-to-Peer (P2P) Videoon- Demand(VoD) Streaming ,The 4th International Conference on Multimedia Computing and Systems (ICMCS'14) ,7, April 14-16 . Marrakesh, Morocco.

[31]Subbarao Lanka, (2014). Optimise UI power using wake-on-approach (Part 1),EE Times-India,4.

[32]Chen Feng, Baochun Li, Network Coding for Content Distribution and Multimedia Streaming in Peer-to-Peer Networks , Department of Electrical and Computer Engineering, University of Toronto.

[33] Luan,H., et al. (2010).Adaptive topology formation for peer-to-peer video streaming, Peer-to-peer Networking and Applications, . 3, p. 186-207.

[34]http://nsl.cs.sfu.ca/projects/nc-svc/nc-svc.tar.gz, 2014.

[35]http://evalsvc.googlecode.com/svn/trunk/jsvm/bin/Output%20svc%20vide os/ , 2014.

[36] JSVM, JSVM Software. Online: CVS Repository of JSVM pserver:jvtuser@garcon.ient. rwth-aachen.de:/cvs/jvt.

[37]http://www.napa-wine.eu/cgi-bin/twiki/view/Public/P2PTVSim, 2 014.

[38] Shabnam Mirshokraie, Mohamed Hefeeda. (2010). Live peer-to-peer streaming with scalable video coding and networking coding, MMSys '10 *In*:Proceedings of the first annual ACM SIGMM conference on Multimedia systems, p. 123- 132.